

# Adaptive Scheduling of Wireless Video Sensor Nodes for Surveillance Applications

Abdallah Makhoul, Rachid Saadi, and Congduc Pham

University of Pau, LIUPPA Laboratory

Avenue de l'Université, BP1155, 64013 Pau Cedex France

abdallah.makhoul@univ-pau.fr, rachid.saadi@univ-pau.fr, congduc.pham@univ-pau.fr

## ABSTRACT

Traditional vision systems for surveillance applications are built essentially from distributed high resolution video cameras and powerful processing units which communicate in general with central servers via a high bandwidth network. Using autonomous and small wireless video sensor nodes can add a much higher level of flexibility, therefore extending the range of surveillance applications that could be considered and, more interestingly, can support dynamic deployment scenario. With such resource-constrained devices power conservation is an important issue but one must also ensure that the application's objectives are met. In this paper, we address the problem of adaptive scheduling of sensor node's activity to extend the network lifetime while fulfilling a critical surveillance application needs. We first present a distributed algorithm that ensures both coverage of the deployment area and network connectivity by providing multiple cover sets to manage Field of View redundancies. Then, we propose a multiple levels activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow for adaptive scheduling. Simulation results are presented to validate the performance of the proposed approach.

## Categories and Subject Descriptors

C.2.4 [Computer Systems Organization]: Computer Communication Networks—*Distributed Systems*

## General Terms

Algorithms

## 1. INTRODUCTION

A Wireless Video Sensor Networks (WVSN) consists of a set of sensor nodes equipped with miniaturized video cameras. In this paper, we are more particularly interested in WVSN for surveillance applications. Surveillance applications have very specific needs due to their inherently critical nature associated to security. For instance, the quality of the captured images or the capture rate must be in adequation with the application's objectives: it is unnecessary

to send an image with a high bandwidth if it is not sufficiently enlightened to allow detection and/or identification of intruders. We can add to these problems the fact that such networks for surveillance strongly lose its interest if the relevant scenes could not be received leading to a bad interpretation of an event which can have disastrous effects.

An important issue is how to ensure and maintain high coverage of the area of interest. The problem of coverage in many-robot system or WSN was largely studied and very interesting results were published. Most of the recent existing works on the connected coverage problem in scalar sensor networks [13, 3, 6, 2] typically assume omnidirectional sensors with disk-like sensing coverage. Thus, two scalar nodes are likely to be redundant if they are close to each other. However, in video sensor networks, video nodes possess limited sensing coverage area (sector coverage) due to the camera constraints and its Field of View (FoV). In addition, two video sensor nodes can be redundant and still be far from each other. Even if we admitted that such a redundancy can be exploited in an energy saving approach, it appears much more difficult to choose the video sensor nodes to put in sleep mode as this is highly dependent from the objectives of the application (several views are often desired) and the environmental conditions in the covered area (far sensors may potentially capture the global scene better due to weather conditions for instance).

One other issue of prime importance is related to energy considerations since the scarcity of energy does have a direct impact on coverage as it is not possible to have all the sensor nodes in activity at the same time. In randomly deployed sensor networks, sensor nodes can be redundant (nodes that monitor the same region) leading to overlaps among the monitored areas. Therefore, a common approach is to define a subset of the deployed nodes to be active while the other nodes can sleep. The result is a schedule of the activity of sensor nodes in such a way that guarantees the deployment area coverage as well as the network connectivity. In addition, it is also desirable to be able to define multiple levels of activity that could correspond to how many samples are captured per unit of time. For instance some surveillance applications may focus on barrier coverage rather than blanket coverage. In this case, sensor nodes at the border of the area of interest should take more samples while interior nodes can decrease their activity. Also, multiple levels of activity may be simply necessary because surveillance applications such as an intrusion detection system has to be able to operate on a long term basis and because one does not know when such intrusions could occur. For video sensor networks efficient scheduling and the ability to provide multiple levels of activity is even more important than in traditional sensing systems (e.g. temperature, pressure,...) as capturing and transmitting images are much more energy-consuming.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PM2HW2N'09, October 26, 2009, Tenerife, Canary Islands, Spain.

Copyright 2009 ACM 978-1-60558-621-2/09/10 ...\$10.00.

This paper presents a framework for adaptively schedule video sensor node's activity while addressing coverage, energy considerations and application's objectives. The final goal is to provide the necessary algorithmic support for a surveillance application to express its objectives. How the objectives are translated into the deployed sensor network or how the application can dynamically configure nodes in the networks are out of the scope of this paper and may need some kind of supervision platforms that are proposed in the recent research literature [7].

Regarding the coverage problem, our contribution is to propose a multiple cover set approach to manage FoV redundancies. Our approach is based on a distributed algorithm that helps each node to organize its neighbors into non-disjoint subsets, each of which being a cover set that overlaps its FoV. Then, based on neighbors activity, a node decides to be active or in sleep mode, without compromising the coverage of its own FoV. Next, taking account the energy-considerations and the application's objectives, the framework also provides multiple levels of activity by providing a default operating mode that links a node's capture rate to the size of its cover sets and two additional operating modes that force a node to be in "high" or "low" status according to the surveillance application's specific objectives. These operating modes uses behavior functions modeled by modified Bezier curves that could define application classes.

The rest of this paper is organized as follows. Section 2 presents the coverage and node scheduling propositions. Section 3 presents the adaptive model that uses behavior functions modeled by modified Bezier curves for providing the necessary flexibility to the end applications. Simulation results are presented in section 4 before the conclusions.

## 2. COVERAGE AND SENSOR NODES SCHEDULING

The coverage problem for wireless video sensor networks can be categorized as:

- *Known-Targets Coverage Problem*, which seeks to determine a subset of connected video nodes that covers a given set of target-locations scattered in a 2D plane.
- *Region-Coverage Problem*, which seeks to find a subset of connected video nodes that ensures the coverage of the entire region of deployment in a 2D plane.

Most of the previous works have considered the known-targets coverage problem [4, 1, 8, 14, 5]. The objective is to ensure at all-time the coverage of some targets with known locations. For example, the authors in [5] organize sensor nodes into mutually exclusive subsets that are activated successively, where the size of each subset is restricted and not all of the targets need to be covered by the sensors in one subset.

In [1], a directional sensor model is proposed, where a sensor is allowed to work in several directions. The idea behind this is to find a minimal set of directions that can cover the maximum number of targets. It is different from the approach described in [4] that aims to find a group of non-disjoint cover sets, each set covering all the targets to maximize the network lifetime.

Concerning the area coverage problem, the existing works focus on finding an efficient deployment pattern so that the average overlapping area of each sensor is bounded. The authors in [9] analyze new deployment strategies for satisfying some given coverage probability requirements with directional sensing models. A model of directed communications is introduced to ensure and repair the network connectivity.

Based on a rotatable directional sensing model, the authors in [12] present a method to deterministically estimate the amount of directional nodes for a given coverage rate. A sensing connected sub-graph accompanied with a convex hull method is introduced to model a directional sensor network into several parts in a distributed manner. With adjustable sensing directions, the coverage algorithm tries to minimize the overlapping sensing area of directional sensors only with local topology information.

Different from the above works, our paper mainly focuses on the area coverage problem and more specifically on scheduling for randomly deployed video sensor nodes. The local coverage objective of each sensor node is to ensure, at all time, the coverage of this area either by itself (if it is active) or by a subset of its neighbors, called the cover set.

### 2.1 Coverage and node's cover set

A video sensor node  $v$  is represented by the FoV of its camera. We consider a 2-D model of a video sensor node where the FoV is defined as a sector denoted by a 4-tuple  $v(P, R_s, \vec{V}, \alpha)$ . Here  $P$  is the position of  $v$ ,  $R_s$  is its sensing range,  $\vec{V}$  is the vector representing the line of sight of the camera's FoV which determines the sensing direction, and  $\alpha$  is the offset angle of the FoV on both sides of  $\vec{V}$ . Figure 1 illustrates the FoV of a video sensor node in our model. In the reminder of this paper, we consider that all video nodes have the same characteristics: same sensing range  $R_s$  and same offset angle  $\alpha$ .

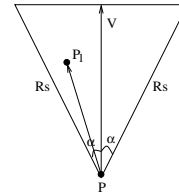


Figure 1: Video Sensing Model

The approach we described in this paper is completely distributed: every video node computes its local solution for coverage. Each node  $v$  covers a sector area thanks to its FoV, which we call  $v$ 's FoV area. Then, its local coverage objective is to ensure, at all time, the coverage of this area either by itself (if it is active) or by a subset of its neighbors. If every video node ensure that its local coverage objective is satisfied then global coverage is also satisfied.

**Definition 1.** We define a cover  $Co_i(v)$  of a video node  $v$  as a subset of video nodes such that:  $\bigcup_{v' \in Co_i(v)} (v'$ 's FoV area) covers  $v$ 's FoV area.

**Definition 2.**  $Co(v)$  is defined as the set of all the covers of node  $v$ .

An example of FoV coverage is shown in Figure 2, where nodes  $v_1, v_2$  and  $v_3$  cover the FoV area of node  $v$ , represented by  $abc$ .

To compute  $Co(v)$ , our model uses four distinctive points:  $a, b, c$  and  $g$  (the center of gravity of  $(abc)$ ) to represent the FoV of  $v$  as shown in Figure 2. Then, we say that  $v$ 's FoV is covered by a set  $Co_i(v) \in Co(v)$  if:

1.  $\forall v' \in Co_i(v)$ ,  $v'$  covers the point  $g$  and at least one of the points  $\{a, b, c\}$
2.  $a, b, c$  and  $g$  are covered by the elements of  $Co_i(v)$ .

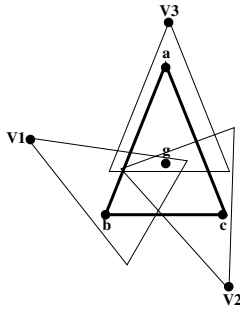


Figure 2: FoV Coverage

In other terms, to compute  $Co(v)$ , a node  $v$  finds the following sets where  $N(v)$  represents the set of neighbors of node  $v$ :

- $A = \{v' \in N(v) : v' \text{ covers point a of the FoV}\}$
- $B = \{v' \in N(v) : v' \text{ covers point b of the FoV}\}$
- $C = \{v' \in N(v) : v' \text{ covers point c of the FoV}\}$
- $G = \{v' \in N(v) : v' \text{ covers point g of the FoV}\}$
- $AG = \{A \cap G\}, BG = \{B \cap G\}, CG = \{C \cap G\}$

Then,  $Co(v)$  is set to the cartesian product of sets  $AG$ ,  $BG$  and  $CG$  ( $\{AG \times BG \times CG\}$ ). Note that the set-intersection function generates  $n + m$  recursive calls in the worst case. Therefore, the intersection of 2 sets can be done with complexity of  $O(n + m)$ , where  $m$  and  $n$  are the cardinality of the two sets respectively. As the size of sets  $A, B, C$  and  $G$  is limited, a video node can easily compute the intersections.

In the example illustrated by Figure 3,  $v$ 's FoV is represented by  $(abcg)$ . To find the set of covers, node  $v$  finds the sets:  $AG = \{v_2, v_3\}$ ,  $BG = \{v_1, v_4\}$  and  $CG = \{v_1, v_5\}$ . Then, following the above method, the set of possible covers for  $v$  is:  $Co(v) = \{\{v\}, \{v_2, v_1\}, \{v_3, v_1\}, \{v_2, v_4, v_5\}, \{v_3, v_4, v_5\}\}$ .

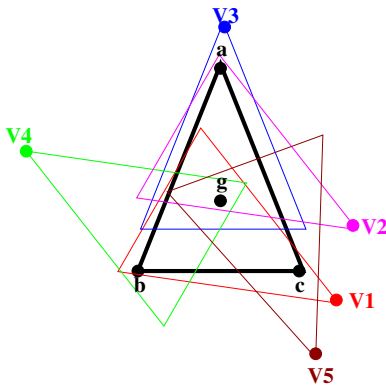


Figure 3: Example of FoV Coverage

This model allows a node  $v$  to construct  $Co(v)$  of its FoV area. Hence, in some cases, it can occur that a cover does not ensure the complete FoV coverage. The effectiveness of such model will be shown in the experimental results section. Note that, one can consider more points such that the midpoints of segments  $[ab]$ ,  $[ac]$  and  $[bc]$ . On the other hand, doing so will reduce the number of covers and consequently increase the number of active nodes. Note that the way coverage is defined could be different. The method we

use in this paper has the advantage of being simple since the main purpose here is to be able to construct cover sets.

## 2.2 Scheduling nodes with cover sets

The scheduling of video sensor nodes operates in 2 phases. The first is a setup phase where each node  $v$  constructs its set of covers  $Co(v)$ . The second phase is the scheduling phase where each node decides to be active or in sleep mode. Our objective is to minimize the number of active nodes while ensuring the whole coverage area.

### 2.2.1 Construction of $Co(v)$

At this step, each video node  $v$  constructs all possible covers ( $Co(v)$ ) that satisfy its local coverage objective (e.g. covering its FoV area). These sets are constructed by considering all the communication graph neighbors. Each node diffuses its position  $P$  and its direction  $\vec{V}$  to its neighbors and receives their informations. Each node constructs the sets  $A, B$  and  $C$  and then computes  $Co_i(v)$  that overlap its FoV and ensure its coverage as explained in section 2.1.

In the literature, most of existing omnidirectional sensing coverage works try to construct disjoint sets of active nodes [13, 3, 6, 2]. In our case, we have the possibility that two or more covers have some video nodes in common. This dependency must be taken into account in the scheduling phase. Hence, selecting one cover also reduces the lifetime of the sensor it has in common with another cover. In this case we can consider the level of energy as a criteria while choosing the active cover. The energy level of the lowest node in  $Co_i(v)$  determines  $Co_i(v)$ 's energy level.

### 2.2.2 Video Node's Scheduling

The activity of video sensor nodes operates in rounds. For each round, every node decides to be active or not based on the activity messages received from its neighbors. Every node orders its sets of covers according to their cardinality, and gives priority to the covers with minimum cardinality. If two sets or more in  $Co$  have same cardinality, priority is then given to the cover with the highest level of energy. Note that another criteria can be defined.

A video node  $v$  receives the activity decisions of its neighbors. Then, it tests if the active nodes belongs to a cover  $Co_i(v)$ . If yes, it goes in sleep mode after sending its decision to its neighbors. In the case where no  $Co_i(v)$  is satisfied, node  $v$  decides to remain in active mode and diffuses its decision.

A video node  $v$  orders the set  $Co(v)$  according to their priorities. Then, it starts with the first cover  $Co_1(v) \in Co(v)$  (which has the lowest cardinality) and tests if it is satisfied. A cover is comprised by a set of video nodes and if one node is or become inactive this cover can not be satisfied.

The node's scheduling process is summarized in Algorithm 1. Node  $v$  receives an activity message from its neighbor  $v'$ , if  $v' \in Co_1(v)$  decided to be inactive, then  $v$  goes to the next cover and so on until it finds an active cover or decides to be active itself. If node  $v' \in Co_1(v)$  is active then node  $v$  check whether all nodes of  $Co_1(v)$  are in active mode in order to go in sleep mode. This process is repeated for each cover and at every round.

## 3. ADAPTATION TO APPLICATION CRITICALITY

For wireless video sensor nodes the frame capture process and transmission to a sink is a huge energy-consuming task. On the other hand, the higher the capture rate is, the better relevant events could be detected and identified. Therefore if we consider a video surveillance application for intrusion detection, video nodes should

---

**Algorithm 1** Scheduling of the node  $v$ 

---

```
1:  $v$  is active
2:  $v$  orders its covers  $Co_i(v) \in Co(v)$   $i = 1, 2, \dots |Co(v)|$ 
3:  $i \leftarrow 1$ 
4: while  $i \leq |Co(v)|$  do
5:    $v$  begins with the cover with highest priority  $Co_i(v)$ 
6:   if neighbor  $v'$  decides to go in sleep mode then
7:     if  $v' \notin Co_i(v)$  then
8:       continue with  $Co_i(v)$ 
9:     end if
10:  else
11:    if  $v' \in Co_i(v)$  then
12:       $v$  chooses the next best priority cover  $Co_{i+1}(v)$ 
13:       $i \leftarrow i + 1$ 
14:    end if
15:  end if
16:  if  $v'$  decides to remain active then
17:    if  $v' \in Co_i(v)$  then
18:      continue with  $Co_i(v)$ 
19:    end if
20:  end if
21:  if  $\forall v', v' \in Co_i(v)$ ,  $v'$  is active then
22:     $v$  becomes inactive and sends its decision to its neighbors
23:  end if
24: end while
25: if no  $Co_i(v)$  is found then
26:    $v$  remains active and sends its decision to its neighbors
27: end if
```

---

capture at a high rate due to the criticality of this type of application. There are other types of non-critical surveillance or monitoring applications where the capture rate does not need to be set to the maximum rate. However, even in the case of an intrusion detection application, it is not realistic to consider that video nodes should always capture at their maximum rate when in active mode because network lifetime is almost as important as coverage in such applications. In general, it is desirable to be able to adjust the capture rate according to the application's requirements. In our approach we express the application criticality by the  $r^0$  variable which can take values between 0 and a defined 1 representing the low and the high criticality level respectively. Low level criticality indicates that the application does not require a high video frame capture rate while a high level criticality does. Then, according to the application's requirements,  $r^0$  could be initialized accordingly into all sensors nodes prior to deployment. It is also possible during the network lifetime to dynamically change  $r^0$ , in all sensor nodes or for a given subset only, if some kind of supervision and control platform is available. Once again, this case is out of the scope of this paper.

### 3.1 Static capture model

A naïve approach would consist in fixing the frame capture rate of all video nodes to a given rate. As illustrated in figure 4, we show how the video nodes capture speed can be regulated proportionally to the critical level  $r^0$ . For instance, high level criticality pushes video nodes to capture at near the maximum frame rate capability. However, this simple approach presents some drawbacks. In fact, (i) setting video nodes to work at full capacity provides very good capture quality but the network lifetime is very short, (ii) although setting the nodes at low capacity saves energy and extends the network lifetime, it provides poor surveillance quality, (iii) choosing a moderate frame capture rate could balance between capture quality

and network lifetime but at the same time sensors can not be fully exploited if it is necessary.

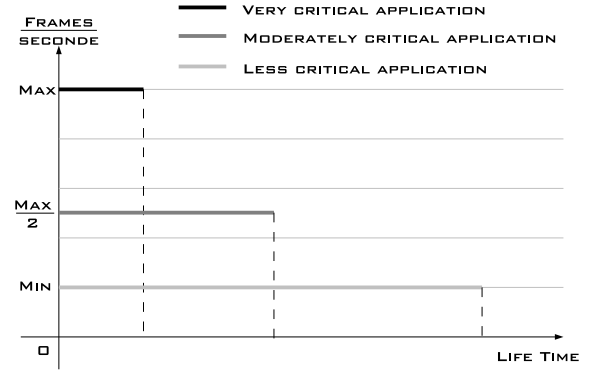


Figure 4: Naïve approach.

### 3.2 Dynamic capture model

To fully exploit the video node capabilities we propose that a video node captures frames at a rate that is defined by the size of its cover set. Obviously, when a node has several covers (see definition 1) it can increase its frame capture rate because if it runs out of energy it can be replaced by one of its covers. On the other hand, the frame capture rate variation should depend on the criticality level of the application as discussed previously. In what follows we will define different application classes which will determine a node's frame capture rate.

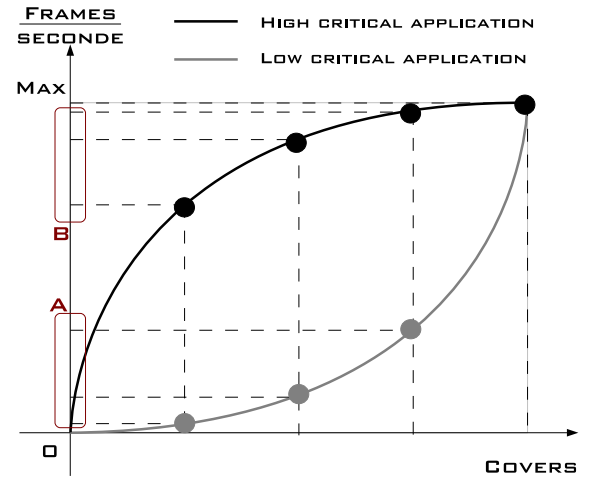


Figure 5: Dynamic approach.

#### 3.2.1 Application classes

We can broadly classify applications into different categories based on their criticality level. In our approach we define two classes of applications: high and low criticality applications. This criticality level is represented by a mathematical function  $y = f_{r^0}(x)$  that we call BV (**BehaVior**) function. This function can oscillate from hyperbolic to parabolic shape as illustrated in Figure 5:

- values on the  $x$  axis are positive integers representing the cardinality of the cover set  $|Co|$ . Integer values lie between

1 and  $max$ , where  $max$  is fixed according to the network topology. A video sensor node that is the only node capable of covering its FoV has a cover size of 1 on the  $x$  axis.

- the  $y$  axis gives the corresponding frame capture rate based on the cardinality of the cover set expressed on the  $x$  axis and the application criticality level ( $r^0$ ).

We now present the contrast between applications that exhibit high and low criticality level in terms of the BV function.

1. **Class 1 "low criticality"**,  $0 \leq r^0 < 0.5$ : this class of applications does not need high frame capture rate. This characteristic is represented by an hyperbolic BV function. As illustrated in figure 5 (box A), most projections of  $x$  values are gathered close to zero (i.e. the majority of the sensors will preserve their energy by capturing slowly).
2. **Class 2 "high criticality"**,  $0.5 \leq r^0 \leq 1$ : This class of applications needs high frame capture rate. This characteristic is represented by a parabolic BV function. As illustrated in figure 5 (box B), most projections of  $x$  values are gathered close to the  $max$  frame capture rate (i.e. the majority of nodes capture at a high rate).

### 3.2.2 The behavior function

We use Bezier curve to model the BV function. Bezier curves are flexible and can plot easily a wide range of geometric curves and have successfully been used for modeling subjectivity in trust evaluation models in pervasive environments [11].

**Definition 3.** The bezier curve is a parametric form to draw a smooth curve. It is fulfilled through some points  $P_0, P_1 \dots P_n$ , starting at  $P_0$  going towards  $P_1 \dots P_{n-1}$  and terminating at  $P_n$ .

In our model we will use a Bezier curve with three points which is called a Quadratic Bezier curve. It is defined as follows:

**Definition 4.** A quadratic Bezier curve is the path traced by the function  $B(t)$ , given points  $P_0, P_1$ , and  $P_2$ .

$$B(t) = (1-t)^2 * P_0 + 2t(1-t) * P_1 + t^2 * P_2. \quad (1)$$

The BV function is expressed by a Bezier curve that passes through three points:

- The origin point ( $P_0(0, 0)$ ).
- The behavior point ( $P_1(b_x, b_y)$ )
- The threshold point ( $P_2(h_x, h_y)$ ) where  $h_x$  represents the highest cover cardinality and  $h_y$  represents the maximum frame capture rate determined by the sensor node hardware capabilities.

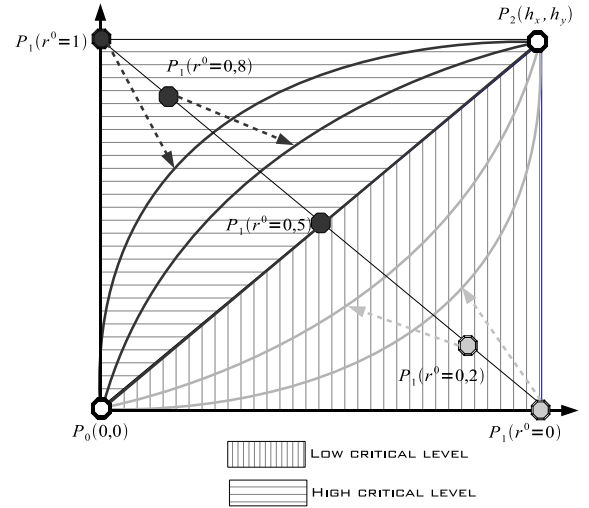
As illustrated in Figure 6, by moving the behavior point  $P_1$  inside the rectangle defined by  $P_0$  and  $P_2$ , we are able to adjust the curvature of the Bezier curve. The BV function describes the application criticality. It takes  $|Co|$  as input on the  $x$  axis and returns the corresponding "frame capture rate" on the  $y$  axis. To apply the BV function with the Bezier curve, we modify this latter to obtain  $y$  as a function of  $x$ , instead of taking a temporal variable  $t$  as input to compute  $x$  and  $y$ . Based on the Bezier curve, let us now define the "BV function":

**Definition 5.** The BV function curve can be drawn through the three points  $P_0(0, 0)$ ,  $P_1(b_x, b_y)$  and  $P_2(h_x, h_y)$  using the Bezier curve as follows:

$$BV : [0, h_x] \longrightarrow [0, h_y] \\ X \longrightarrow Y$$

$$BV_{P_1, P_2}(X) = \begin{cases} \frac{(h_y - 2b_y)}{4b_x^2} X^2 + \frac{b_y}{b_x} X & \text{if } (h_x - 2b_x = 0) \\ (h_y - 2b_y)(\alpha(X))^2 + 2b_y \alpha(X), & \text{if } (h_x - 2b_x \neq 0) \end{cases}$$

$$\text{Where } \alpha(X) = \frac{-b_x + \sqrt{b_x^2 - 2b_x * X + h_x * X}}{h_x - 2b_x} \wedge \begin{cases} 0 \leq b_x \leq h_x \\ 0 \leq X \leq h_x \\ h_x > 0 \end{cases} \quad (2)$$



**Figure 6: The Behavior curve functions**

### 3.2.3 The criticality level $r^0$

As discussed above, the criticality level  $r^0$  of an application is given into the interval  $[0, 1]$ . According to this level, we define the criticality function called  $Cr$  which operates on the behavior point  $P_1$  to control the BV function curvature.

According to the position of point  $P_1$  the Bezier curve will morph between parabolic and hyperbolic form. As illustrated in figure 6 the first and the last points delimit the curve frame. This frame is a rectangle and is defined by the source point  $P_0(0, 0)$  and the threshold point  $P_2(h_x, h_y)$ . The middle point  $P_1(b_x, b_y)$  controls the application criticality. We assume that this point can move through the second diagonal of the defined rectangle  $b_x = \frac{-h_y}{h_x} * b_y + h_x$ .

We define the  $Cr$  function as follows, such that varying  $r^0$  between 0 and 1 gives updated positions for  $P_1$ :

$$Cr : [0, 1] \longrightarrow [0, h_x] * [0, h_y] \\ r^0 \longrightarrow (b_x, b_y) \\ Cr(r^0) = \begin{cases} b_x = -h_x \times r^0 + h_x \\ b_y = h_y \times r^0 \end{cases} \quad (3)$$

Level  $r^0$  is represented by the position of point  $P_1$ . If  $r^0 = 0$   $P_1$  will have the coordinate  $(h_x, 0)$ . If  $r^0 = 1$   $P_1$  will have the coordinate  $(0, h_y)$ .

## 4. EXPERIMENTAL RESULTS

To evaluate our approach we conducted a series of simulations based on the discrete event simulator OMNet++ [10]. The results were obtained from iterations with various densities on a  $100m \times 100m$  area. Nodes have equal communication and sensing ranges of  $30m$  and  $25m$  respectively, an offset angle  $\alpha$  of  $\pi/6$ , a battery life of 100 units, random position  $P$  and random direction  $\vec{V}$ . A simulation starts by a neighborhood discovery. Each node gathers positions and directions of its neighbors and finds the sets  $AG$ ,  $BG$  and  $CG$ . Then, round by round each node decides to be active or not. For these simulations, the round duration is set to 1s. Simulation ends as soon as the subset of nodes with power left is disconnected (when all active nodes have no neighbors anymore). We run each simulation 15 times to reduce the impact of randomness. Then, proportion of active nodes, area coverage and criticality impact were observed and analyzed.

### 4.1 Coverage results

In this section, we focus on the coverage results. We varied the deployed nodes density from 50 to 200 nodes in a  $100m \times 100m$  area, and we noted at each round the percentage of active nodes and the percentage of area coverage. At the end of a round an active node decreases its battery life by one unit.

#### 4.1.1 Percentage of active nodes

The percentage of active nodes represents the average number of nodes involved in the active set over the initial number of deployed sensors. This metric reflects, to a certain extent, the effectiveness of the proposed scheduling approach. Figure 7 shows the evolution of this ratio for each round during the network lifetime.

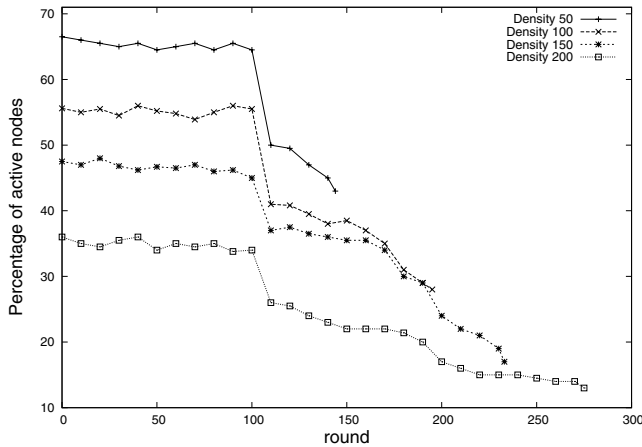


Figure 7: Percentage of active nodes

Before the first drop at round 100, which corresponds to the extinction of a subset of nodes (the nodes that initially did not have redundant cover sets), the number of active nodes varies from less than 67%, for density 50, to 36% for density 200. We notice that increasing the sensing range decreases the percentage of active nodes at each round and increases the network lifetime.

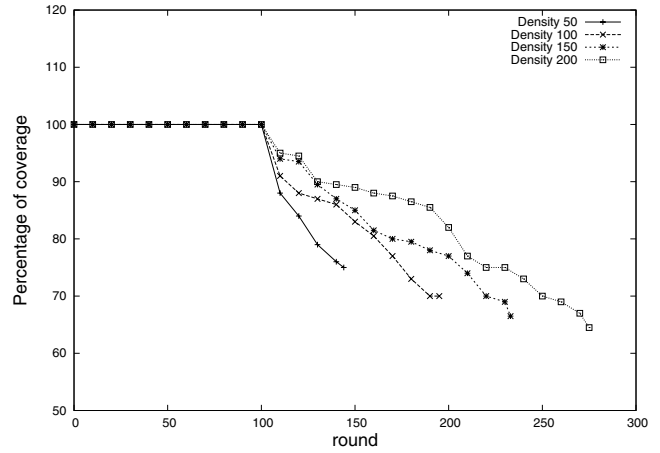


Figure 8: Percentage of coverage.

#### 4.1.2 Percentage of Area Coverage

The main objective of our approach is to maintain a full area coverage at each round. We define the full area coverage as the region covered initially by the whole network (i.e when all the deployed nodes are active). This area represents the union of all FoV areas of the deployed nodes. Figure 8 shows the percentage of area coverage round by round. This percentage is the ratio between the area covered by the set of active nodes over the initial covered area.

The initial sensing coverage is preserved for 100 rounds which is equal to a node's lifetime. At round 100, as expected, a set of nodes run out of energy. We can observe that in all the cases our algorithm guarantees a sensing coverage of at least 67% of the deployment area.

### 4.2 Criticality impact

For these set of experiments, we deploy randomly 150 sensor nodes in a  $75m \times 75m$  area. Each sensor node captures with a given number of frames per second (between 0fps and 6fps) in which case the battery capacity is decreased accordingly by 1 unit per captured frame (initial battery capacity is set to 100 units). The performance evaluation was realized with 2 fixed frame capture rate scenarios (3fps and 6fps) and 3 levels of application criticality that follows the dynamic rate model:  $r^0 = 0$  (low criticality),  $r^0 = 0.5$  and  $r^0 = 1$  (high criticality) where a node's frame capture rate depends on the size of its cover set.

Figure 9 shows the percentage of coverage while varying the frame capture rate. Figure 10 shows the average frame capture rate of all active nodes per round which is representative of the surveillance application quality during the network lifetime.

As the application criticality  $r^0$  is varied the frame capture rate of each sensor node that depends on its cover set size is modified according to the behavior function. Beyond the fact that our multiple levels activity model allows an application to specify a criticality level according to its objectives, the model also optimizes the area coverage as well as the network lifetime. For instance, we can see in figure 10 that the high criticality scenario gives a mean frame capture rate of 4.63fps which should draw in figure 9 a curve that lies between the 6fps and the 3fps curves. However, figure 9 shows a high criticality curve very close to the 3fps curve which indicates that our dynamic model can provide the same percentage of coverage but at a higher frame capture rate which should mean a better surveillance quality.

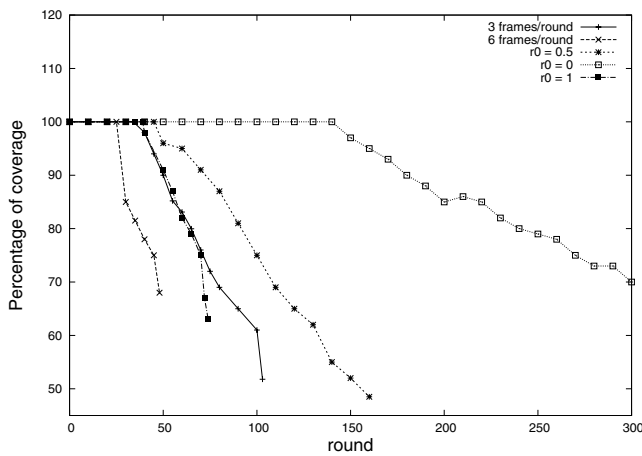


Figure 9: Percentage of coverage while varying the frame capture rate.

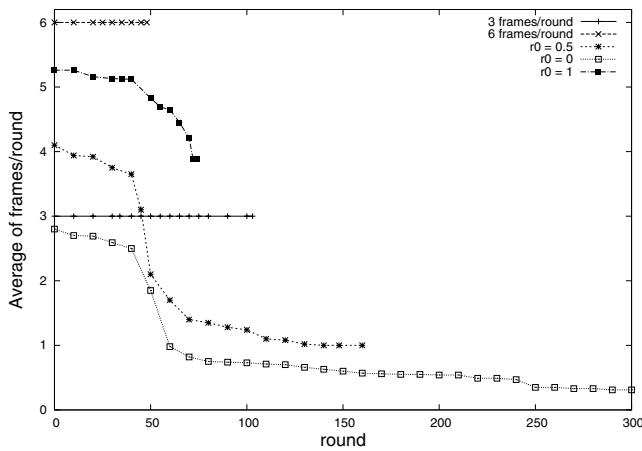


Figure 10: Average frame capture rate per round.

## 5. CONCLUSION

This paper addresses the problem of scheduling video sensor nodes for critical surveillance applications. Based on a coverage model that handle FoV redundancies by providing multiple covers per sensor node, we proposed a multiple levels activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow for adaptive scheduling. Our contribution is to provide the necessary algorithmic support for a surveillance application to express its objectives in terms of criticality level. Besides providing a model for translating a subjective criticality level into a quantitative parameter of the surveillance system, our proposed approach for video sensor nodes can also optimize the resource usage by dynamically adjusting the provided service level. Future works will investigate how the risk factor in surveillance systems can be handle by our multiple levels activity model in order to automatically configure sensor nodes according to the environment stimulus.

## 6. REFERENCES

- [1] J. Ai and A. A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1):21–41, 2006.
- [2] J. Bahi, A. Makhoul, and A. Mostefaoui. Improving lifetime and coverage through a mobile beacon for high density sensor networks. *The Second IEEE International Conference on Sensor Technologies and Applications, SENSORCOMM08*, pages 335–341, august 2008.
- [3] J. Bahi, A. Makhoul, and A. Mostefaoui. Hilbert mobile beacon for localization and coverage in sensor networks. *International Journal of Systems Science, Taylor & Francis*, 39(11):1081–1094, Novembre 2008.
- [4] Y. Cai, W. Lou, M. Li, and X.-Y. Li. Target-oriented scheduling in directional sensor networks. *26th IEEE International Conference on Computer Communications, INFOCOM 2007*, pages 1550–1558, 2007.
- [5] M. X. Cheng, L. Ruan, and W. Wu. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. in *IEEE INFOCOM*, 2005.
- [6] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic. Localized sensor area coverage with low communication overhead. *IEEE Transactions on Mobile Computing (TMC)*, 7(5):661–672, 2008.
- [7] S. Laplace, M. Dalmau, and P. Roose. Kalinahia: Considering quality of service to design and execute distributed multimedia applications. *Proceedings of the IEEE/IFIP Int'l Conference on Network Management and Management Symposium*, 2008.
- [8] H. Liu, P. Wan, and X. Jia. Maximal lifetime scheduling for sensor surveillance systems with k sensors to one target. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1526–1536, 2006.
- [9] H. Ma and Y. Liu. Some problems of directional sensor networks. *International Journal of Sensor Networks*, 2(1-2):44–52, 2007.
- [10] OMNeT++. <http://www.omnetpp.org/>.
- [11] R. Saadi, O. Hasan, J. Pierson, and L. Brunie. Establishing trust beliefs based on a uniform disposition to trust. *IEEE international Conference on Signal Image and Technologie Based Systems*, 2007.
- [12] D. Tao, H. Ma, and L. Liu. Coverage-enhancing algorithm for directional sensor networks. *Lecture Notes in Computer Science - Springer*, pages 256–267, November 2006.
- [13] C. Wang, M. T. Thai, Y. Li, F. Wang, and W. Wu. Coverage based information retrieval for lifetime maximization in sensor networks. In *IEEE GlobeCom*, Nov. 2007.
- [14] J. Wang, C. Niu, , and R. Shen. Randomized approach for target coverage scheduling in directional sensor network. *Lecture Notes in Computer Science - Springer*, pages 379–390, 2007.